IN THE CLAIMS

1. (Currently Amended) In a digital communications network having network cards, a method comprising:

controlling applications executed within the network, wherein controlling the applications comprises,

transitioning each of the applications between one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the no-provisioning state.

2. (Original) The method of claim 1, wherein an application state machine controls the execution of the application.

3. (Original) The method of claim 2, further comprising:

receiving control messages from a shelf manager; and

communicating via APIs to the application, wherein the shelf manager may be located on a remote network card.

4. (Canceled)

5. (Currently Amended) The method of claim 1, wherein the standby states comprise~~:~~

~~a standby ready state, and~~

a standby locked state.

6. (Currently Amended) In a digital communications network having network cards, a method comprising:

switching the state of an application in an active state to a standby state, comprising,

transitioning the application from the active state to a quiescent state on an active card of the network cards; and

transitioning the application from the quiescent state to the standby state on a standby card of the network cards,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of the commands required by each of the applications are loaded into a memory of the standby card for executing each of the applications during the standby state.

7. (Currently Amended) In a digital communications network having network cards, a method comprising:

upgrading code of an application in an active state on an active card of the network cards to a standby locked state on a standby card of the network cards comprising,

transitioning the application from the active state to a no provisioning state during which at least a portion of network management requests for configuring the active network card are rejected;

transitioning the application from the no provisioning state to a quiescent state during which memories of the active card and the standby card are synchronized during the quiescent state; and

transitioning the application from the quiescent state to the standby locked state during which an application is in a ready state in the standby card but does not communicate with the corresponding application of the active card.

8. (Original) The method of claim 7, wherein the standby locked state does not allow disk database access nor access to write to RAM.

9. (Original) The method of claim 7, wherein the no provisioning state does not allow access to write to a disk database.

10. (Original) The method of claim 7, wherein the quiescent state does not allow access to write to a disk database nor access to write to RAM.

11. (Currently Amended) In a digital communications network having network cards, a method comprising:

upgrading code of an application in an standby state to an active state comprising,

transitioning the application from the standby state on a standby card of the network cards to a no provisioning state on an active card of the network cards; and

transitioning the application from the no provisioning state to the active state, wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state, and wherein at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state.

12. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for controlling applications executed within the network, wherein the means for controlling the applications comprises,

means for transitioning each of the applications between one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the no-provisioning state.

13. (Original) The system of claim 12, further comprising:

means for receiving control messages from a shelf manager; and

means for communicating via APIs to the application, wherein the shelf manager may be located on a remote network card.

14. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for switching the state of an application in an active state to a standby state, comprising,

means for transitioning the application from the active state to a quiescent state on an active card of the network cards; and

means for transitioning the application from the quiescent state to the standby state on a standby card of the network cards,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active state;

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of the commands required by each of the applications are loaded into a memory of the standby card for executing each of the applications during the standby state.

15. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for upgrading code of an application in an active state on an active card of the network cards to a standby locked state on a standby card of the network cards comprising,

means for transitioning the application from the active state to a no provisioning state during which at least a portion of network management requests for configuring the active network card are rejected;

means for transitioning the application from the no provisioning state to a quiescent state during which memories of the active card and the standby card are synchronized during the quiescent state; and

means for transitioning the application from the quiescent state to the standby locked state during which an application is in a ready state in the standby card but does not communicate with the corresponding application of the active card.

16. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for upgrading code of an application in an standby state to an active state comprising,

means for transitioning the application from the standby state on a standby card of the network cards to a no provisioning state on an active card of the network cards; and

means for transitioning the application from the no provisioning state to the active state, wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state, and wherein at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state.

17. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

controlling applications executed within the network, wherein controlling the applications comprises,

transitioning each of the applications between one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the no-provisioning state.

18. (Original) The computer-readable medium of claim 17 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:

receiving control messages from a shelf manager; and

means for communicating via APIs to the application, wherein the shelf manager may be located on a remote network card.

19. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

switching the state of an application in an active state to a standby state, comprising,

transitioning the application from the active state to a quiescent state on an active card of the network cards; and

transitioning the application from the quiescent state to the standby state on a standby card of the network cards,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of the commands required by each of the applications are loaded into a memory of the standby card for executing each of the applications during the standby state.

20. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

upgrading code of an application in an active state to a standby locked state comprising,

transitioning the application from the active state to a no provisioning state on an active card of the network cards during which at least a portion of network management requests for configuring the active network card are rejected;

transitioning the application from the no provisioning state to a quiescent state during which memories of the active card and the standby card are synchronized during the quiescent state; and

transitioning the application from the quiescent state to the standby locked state on a standby card of the network cards during which an application is in a ready state in the standby card but does not communicate with the corresponding application of the active card.

21. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

upgrading code of an application in an standby state to an active state comprising,

transitioning the application from the standby state on a standby card of the network cards to a no provisioning state on an active card of the network cards; and

transitioning the application from the no provisioning state to the active state, wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state, and wherein at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state.

22. (Currently Amended) In a digital communications network, a system for controlling tasks performed on network cards comprising:

a CPU subsystem;

one or more input/output ports connected to the CPU subsystem for communicating with the network; and

special hardware connected to the CPU subsystem via a bus, wherein the CPU subsystem controls applications executed within the network that transition from one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state,

wherein substantially all commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the no-provisioning state.

23. (Original) The system of claim 22 further comprising a disk database connected to the CPU subsystem via a PCI bus.

24. (Original) The system of claim 22, wherein the CPU subsystem comprises:

a central processing unit;

a system controller connected to the central processing unit;

random access memory connected to the system controller; and

an application state machine for transitioning applications between one of a plurality of active states and one of a plurality of standby states.